BAB II
LANDASAN TEORI

2.1 Rancang Bangun

Menurut Pressman (2010, p.68) menjelaskan bahwa rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan.

2.2 Aplikasi

Menurut Safaat (2012, p.9) perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan Open Office.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.3 Arisan

Arisan adalah kegiatan mengumpulkan uang atau barang yang bernilai sama oleh beberapa orang kemudian diundi di antara mereka untuk menentukan siapa yang memperolehnya, undian dilaksanakan dalam sebuah pertemuan secara

berkala sampai semua anggota memperolehnya. Sumber : http://kbbi.web.id/arisan Diakses Pada Pukul 22:20 Tanggal 3 November 2016.

2.4 Android

Safaat (2012, p.15) mendefinisikan Android adalah suatu sistem operasi yang didesain sebagai *platform open source* untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android menyediakan semua *tools* dan *framework* untuk mengembangkan aplikasi dengan mudah dan cepat. Dengan adanya Android SDK (*Software Developement Kit*) pengembang aplikasi dapat memulai pembuatan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

Pada tahun 2005, Google membeli Android, Inc. yang merupakan pendatang baru yang mengembangkan *software* untuk *smartphone*. Pada saat itu dunia mengira bahwa Google akan memproduksi *smartphone*. Anggapan itu ternyata salah karena Google menyatakan bahwa ambisi Android bukan hanya untuk mengembangkan sebuah ponsel melainkan suatu *platform* yang dapat digunakan di banyak ponsel dan perangkat lainnya. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, sebuah grup nonprofit yang terdiri dari beberapa perusahaan *hardware, software*, telekomunikasi dan perusahaan lainnya termasuk Google, HTC, Intel Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android tanggal 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi *software* dan *open platform* perangkat seluler.

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Sekitar September 2007 Google mengenalkan Nexus One, *smartphone* yang menggunakan sistem operasi Android versi 1.0. Ponsel ini diproduksi oleh HTC Corporation dan mulai dipasarkan pada 5 Januari 2008. Pada tahun 2009

perkembangan perangkat berbasis Android semakin pesat. Lebih dari 20 jenis perangkat mobile menggunakan sistem operasi Android. Versi Android yang dirilis pada tahun 2009 antara lain Cupcake(1.5), Donut(1.6), dan Eclair (2.0 dan 2.1). Hal ini didukung dengan lebih dari 20 *gadget* yang menggunakan versi tersebut.

Frozen Yogurt Versi Android 2.2 dirilis pada tahun 2010. Pada tahun 2010 lebih dari 60 perangkat menggunakan Android sebagai sistem operasinya. Pada saat itu Android menjadi platform ponsel dengan penjualan terbaik kedua setelah Blackberry. Samsung Galaxy Tab GT-P1000 adalah salah satu smartphone/tabletp pc dengan sistem operasi Android versi 2.2. Daftar versi-versi Android yang dapat dilihat pada tabel 2.1.

Tabel 2.1 Versi – versi Android

Versi Android	**API Level**	**Nickname**
Android 1.0	1	
Android 1.1	2	
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Eclair
Android 2.01	6	Eclair
Android 2.1	7	Eclair
Android 2.2	8	Froyo (Frozen Yogurt)
Android 2.3	9	Gingerbread
Android 2.3.3	10	Gingerbread
Android 3.0	11	Honeycomb
Android 3.1	12	Honeycomb
Android 3.2	13	Honeycomb
Android 4.0	14	Ice Cream Sandwich
Android 4.0.3	15	Ice Cream Sandwich
Android 4.1	16	Jelly Bean

Dari Tabel 2.1 dapat dilihat pada setiap perubahan versi android terdapat perubahan *Application Programming Interface* (API) Level. API Level adalah nilai yang menunjukkan revisi framework pada platform Android. Nomor versi android yang terus berubah dikarenakan perubahan API Level. Setiap versi android mendukung tepat satu API Level, namun tetap mendukung API Level sebelumnya. API Level akan menentukan apakah suatu aplikasi bisa dijalankan pada suatu *platform* Android atau tidak.

Pada saat perilisan perdana Android tahun 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat selular. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Pada masa saat ini kebanyakan vendor – vendor smartphone sudah memproduksi smartphone berbasis Android, vendor – vendor itu antara lain Motorola, Samsung, Huawei , Archos, Platformstation Camangi Dell, Nexus, SciPhone,Wayteq dan masih banyak lagi vendor smart phone di dunia ini yang memproduksi android. Hal ini karena android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun. Android sebagai platform masa depan karena :

1. Terbuka (*Open Source Platform*): *Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
2. *Free*, artinya Android adalah *platform* yang bebas untuk develope. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

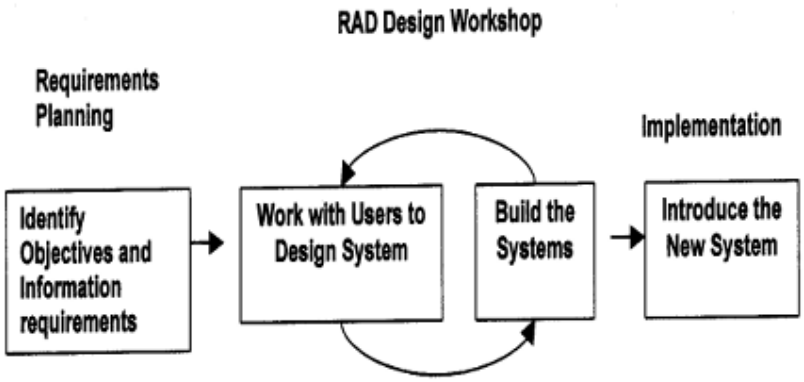
2.5 *Android Application Package* (APK)

Mulyadi (2010, p.52) *Android Application Package* adalah paket aplikasi android yang umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat android. APK pada dasarnya seperti zip file, karena terdiri dari kumpulan file. APK dapat diperoleh dari berbagai metode, seperti menginstal sebuah aplikasi melalui *play store, download* dari sebuah situs web atau membuat sendiri dengan bahasa java. Jika memiliki file APK pada komputer dan ingin menginstalnya pada *handphone,* maka dapat menjalankan perintah ‘adb install apkname.apk’ untuk install aplikasi melalui USB ke *handphone.*

2.6 Metode Pengembangan Perangkat Lunak Rapid Application Development (RAD)

Metode penelitian yang digunakan adalah Pengembangan Perangkat Lunak Menggunakan Metode Rapid Application Development (RAD). Menurut Sutanto (2004, p.353) Rapid Application Development (RAD) adalah pengembangan dari beberapa metode atau teknik terstruktur khususnya dalam pengolahan data untuk menghasilkan informasi, misalnya dengan mengintegrasikan metode *Prototyping*, metode SDLC dan teknik *Joint Apllication* Development untuk mempercepat pengembangan sistem informasi.

Metode Rapid Application Development (RAD) memiliki tiga faktor utama yaitu: kelompok pemakai sistem harus memiliki staf senior yang benar-benar berdedikasi terhadap pengembangan sistem informasi yang memudahkan mereka dalam berhubungan dengan pengembangan sistem, tim pengembang sistem harus stabil dan memiliki kemampuan yang memadai, dan lingkup aplikasi harus komersial dengan penentuan-penentuan permintaan yang jelas dari sekelompok pemakai sistem. Model pengembangan perangkat lunak Rapid Application Development (RAD) disajikan pada Gambar 2.1.



Gambar 2.1 Model Rapid Application Development (RAD)

Rapid Application Development (RAD) memiliki tahap-tahap sebagai berikut:

1. Perencanaan Syarat-Syarat (*Requirement Planning*)

   Pada tahap ini, user dan analis melakukan semacam pertemuan untuk melakukan identifikasi tujuan aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan. Pada tahap ini hal terpenting adalah adanya keterlibatan dari kedua belah pihak, bukan hanya sekedar persetujuan akan proposal yang sudah dibuat. Untuk lebih jauh lagi, keterlibatan user bukan hanya dari satu tingkatan pada suatu organisasi, melainkan beberapa tingkatan organisasi sehingga informasi yang dibutuhkan untuk masing-masing *user* dapat terpenuhi dengan baik.

2. Proses Desain (*Design Workshop*)

   Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara user dan *analyst*. Untuk tahap ini maka keaktifan user yang terlibat sangat menentukan untuk mencapai tujuan, karena user bisa langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain.

3. Implementasi (*Implementation*)

   Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh *user* dan *analyst*, maka pada tahap ini programmer mengembangkan

desain menjadi suatu program. Setelah program selesai baik itu sebagian maupun secara keseluruhan, maka dilakukan proses pengujian terhadap program tersebut apakah terdapat kesalahan atau tidak sebelum diaplikasikan pada suatu organisasi. Pada saat ini maka user bisa memberikan tanggapan akan sistem yang sudah dibuat serta persetujuan mengenai sistem tersebut. Adapun hal terpenting adalah bahwa keterlibatan user sangat diperlukan supaya sistem yang dikembangkan dapat memberikan kepuasan kepada user, dan di samping itu, sistem yang lama tidak perlu dijalankan secara paralel dengan sistem yang baru.

2.7 *Unified Modeling Language* (UML)

Nugroho (2010, p.6) Mengatakan UML (Unified Modeling Languange) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’ pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. UML merupakan suatu kesatuan dari bahasa pemodelan yang telah dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan Object *Oriented Software Engineering* (OOSE). Metode Booch ini dikenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan tahapan proses analisis serta *design* kedalam 4 tahapan iterative, yakni identifikasi pada kelas dan obyek, identifikasi semantik dari hubungan kelas dan obyek tersebut, selanjutnya perinciaan *interface* dan yang terakhir adalah implementasi. Pemodelan OMT yang telah dikembangkan Rumbaugh didasarkan pada sebuah analisis terstruktur dan pemodelan *entity-relationship*

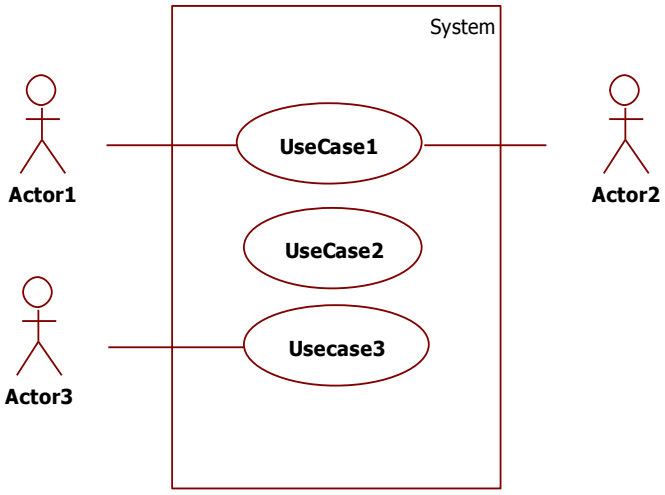
2.7.1 *Use Case Diagram*

Menurut Widodo (2011, p.10) *use case* bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Berikut komponen yang terdapat pada *use case diagram* disajikan pada tabel 2.2.

Tabel 2.2 Komponen *use case diagram*

Nama Komponen	Keterangan	Simbol
Use Case	*Use Case* digunakan untuk menjelaskan apa yang dilakukan oleh aktor dan sistem.	
Actor	*Actor* adalah segala sesuatu yang berinteraksi dengan sistem.	
Association	*Association* digunakan untuk menggambarkan hubungan *Actor* dengan *use case*.	

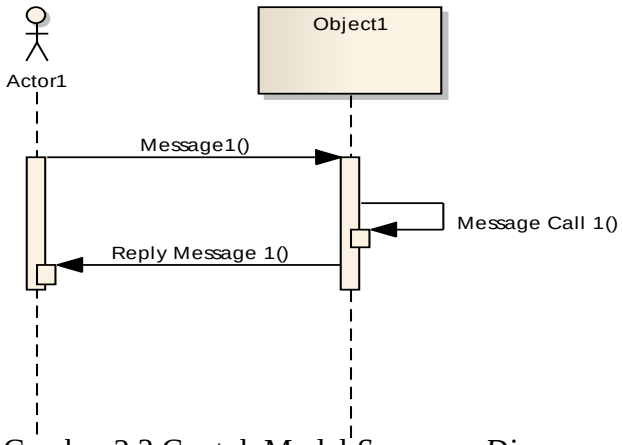
Dalam sebuah *use case*, pengguna biasa disebut dengan actor. Actor memiliki peran yang dimainkan oleh pengguna dalam berinteraksi dengan sistem. *Use case* sebagai alat bantu terbaik berfungsi menstimulasi pengguna potensial yang menggambarkan suatu sistem dari sudut pandangnya. Diagram *use case* memiliki 3 notasi yang menunjukkan aspek dari sistem tersebut. Berikut contoh model *use case* diagram disajikan pada gambar 2.2.



Gambar 2.2 Contoh model *use case diagram*

2.7.2 *Sequence Diagram*

Menurut Widodo (2011, p.10) Secuence diagram bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Pada diagram ini ditunjukkan sejumlah contoh obyek dan pesan yang diletakkan diantara beberapa obyek dalam use case. Komponen utama dari *sequence diagram* yakni obyek yang dituliskan dengan kotak segiempat bernama, pesan diwakili dengan garis tanda panah dan waktu ditunjukkan dengan *progress vertical*. Berikut contoh model *Sequence Diagram* disajikan pada gambar 2.3.



Gambar 2.3 Contoh Model *Sequence Diagram*

Komponen yang terdapat pada *Sequence Diagram* disajikan pada tabel 2.3

Tabel 2.3 Komponen *Sequence Diagram*

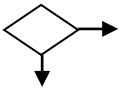
Nama Komponen	Keterangan	Simbol
Object	Merupakan simbol yang digunakan untuk menggambarkan objek yang berinteraksi.	
Object lifeline	*Object lifeline* adalah simbol yang digunakan untuk menggambarkan masa hidup suatu objek.	
Activations	*Activations* adalah Periode dimana objek sedang beroperasi.	
Nama Komponen	**Keterangan**	**Simbol**

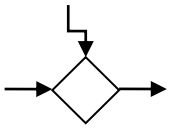
Message	*Message* adalah pesan yang disampaikan dari satu objek ke objek lain.	

2.7.3 *Activity Diagram*

Menurut Satzinger et al. (2010, p.141) *Activity diagram* merupakan sebuah tipe dari diagram *workflow* yang menggambarkan tentang aktivitas dari pengguna ketika melakukan setiap kegiatan dan aliran sekuensial. Berikut komponen yang terdapat pada *activity diagram* disajikan pada tabel 2.4.

Tabel 2.4 Komponen *Activity Diagram.*

Nama Komponen	Keterangan	Simbol
Activity	*Activity* adalah simbol yang digunakan utuk menggambarkan suatu aktivitas atau kegiatan yang dilakukan.	
Initial	*Initial* adalah simbol yang digunakan untuk memulai suatu aur *activity diagram.*	
Final	*Final* adalah simbol yang digunakan untuk mengakhiri suatu alur *activity diagram.*	
Decision	*Decision* adalah simbol yang digunakan menggambarkan pilihan atau keputusan	

Merge Point	*Merge Point* adalah simbol yang sama dengan *decision* namun digunakan untuk menggambarkan titik penggabungan beberapa alur menjadi satu.	

2.8 HTML 5

James (2012, p.14) mendefinisikan "HTML 5 adalah spesifikasi atau aturan baku yang didalamnya sudah terintegrasi teknologi *Application Programming Interface* (API) yang dapat membantu aplikasi web berkomunikasi".

2.9 NoSql Database dan *Firebase*

Sadelage.J.Pramond, Fowler Martin (2012, p.7) *Firebase* adalah peyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disingkronisasikan untuk *client* yang berbeda – beda dan disimpan pada *cloud*-nya *Firebase*. Istilah *NoSql* mengacu pada sekumpulan database yang tidak mengikuti aturan sebagaimana relational *database* yang menggunakan *schema*, baris kolom untuk merepresentasikan data serta tidak menggunakan *Structured Query Language (SQL)* untuk memanipulasi data-data didalamnya. Meskipun istilah *NoSql* pertama kali tercatat sebagai nama sebuah aplikasi *open source relational database* Strozzi NoSql yang dikembangkan Carlo Strozzi pada sekitar akhir tahun 1990. Istilah *NoSql* menjadi dikenal setelah acara konferensi *(Developer Meet Up)* pada 11 Juni 2009 di San Francisco yang diselnggarakan oleh Johan Oskarsson, Seorang *Software Developer* yang berbasis di London .

NoSql meliputi berbagai macam teknologi *database* yang dikembangkan untuk menjawab kebutuhan yang terkait dengan pengembangan aplikasi modern, yang tidak dapat atau sulit dilakukan menggunakan relational *database*, seperti:

a) Kebutuhan aplikasi untuk dapat menyimpan dan mengolah berbagai macam data dengan jumlah transaksi dan volume data yang besar serta pertumbuhan data yang sangat pesat mencakup *structured, semi-structure, unstructure dan polymorphic data*.

b) Kebutuhan untuk mendeliver produk secara cepat, menyesuaikan dengan proses pengmbangan yang dilakukan secara *agile* dengan interasi antara satu hingga dua minggu.
c) Kebutuhan untuk membangun aplikasi yang dapat diakses oleh banyak pengguna dari berbagai macam perangkat.
d) Kebutuhan akan penggunaan teknologi *open source* dan arsitektur berbasis *cloud* dengan skema harga yang lebih terjangkau.

Beberapa kelebihan menggunakan *NoSql* adalah sebagai berikut:

a) Dapat memproses volume data dan transaksi yang besar mencakup *structured, semi-sctuctured, unstructured data*.
b) Mendukung mekanisme pengembangan secara *Object-oriented* yang mudahd digunakan dan fleksibel.
c) Menggunakan arsitektur yang terdistribusi sehingga mudah *discale-out* secara geografis.

2.10 Wawancara

Menurut Esterberg dalam Sugiyono (2013, p.231) wawancara merupakan pertemuan dua orang untuk bertukar informasi dan ide melalui tanya jawab, sehingga dapat dikontruksikan makna dalam suatu topik tertentu..

a. Wawancara Terstruktur

Wawancara terstruktur digunakan sebagai teknik pengumpulan data, bila peneliti atau pengumpul data telah mengetahui dengan pasti tentang informasi apa yang akan diperoleh. Oleh karena itu dalam melakukan wawancara, pengumpul data telah menyiapkan instrumen penelitian berupa pertanyaan pertanyaan tertulis yang alternatif jawabannya pun telah disiapkan.

b. Wawancara Tidak Terstruktur

wawancara tidak terstruktur adalah wawancara yang bebas dimana peneliti tidak menggunakan pedoman wawancara yang telah tersusun secara sistematis dan lengkap untuk pengumpulan datanya. Pedoman wawancara yang digunakannya berupa garis-garis besar permasalahan yang akan ditanyakan.